

**FIȘA DISCIPLINEI**  
**2024-2025**  
**Anul II/Semestrul II**

**1. Date despre program**

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918” din Alba Iulia
1.2. Facultatea	de Informatică și Inginerie
1.3. Departamentul	Departamentul de Informatică, Matematică și Electronică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii/Calificarea	ESCO-08: 2511/ Systems Analyst, 2512/ Software developers Informatică/ Administrator baze de date - 252101; Administrator de rețea de calculatoare - 252301; Analist - 251201; Asistent de cercetare în informatică - 214918; Profesor în învățământul gimnazial - 233002; Programator - 251202; Proiectant sisteme informatice - 251101

**2. Date despre disciplină**

2.1. Denumirea disciplinei	<b>Tehnici avansate de programare</b>		2.2. Cod disciplină	INFO213			
2.3. Titularul activității de curs	Kadar Manuella						
2.4. Titularul activității de seminar	Domșa Ovidiu						
2.5. Anul de studiu	II	2.6. Semestrul	II	2.7. Tipul de evaluare (E/C/VP)	C	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	O

**3. Timpul total estimat**

3.1. Numar ore pe saptamana	4	din care: 3.2. curs	2	3.3. seminar/laborator	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6. seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					8
Tutoriat					-
Examinări					2
Alte activități .....					-

3.7 Total ore studiu individual (a+b+c)	44
3.8.Total ore activități universitare (d+e+f+3.4)	56
3.9 Total ore pe semestru (3.7+3.8)	100
3.10 Numărul de credite (1ECTS=25)	4

**4. Precondiții (acolo unde este cazul)**

4.1. de curriculum	Discipline de parcurs din semestrele anterioare, ex: <b>1. Programare orientată pe obiecte</b>
4.2. de competențe	Competențele oferite de disciplinele enumerate mai sus, <b>Programarea în limbaje de nivel înalt</b> <ul style="list-style-type: none"> <li>• Descrierea adecvată a - paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</li> <li>• Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</li> <li>• Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.</li> </ul>

**5. Condiții (acolo unde este cazul)**

5.1. de desfășurare a cursului	Sala dotată cu videoproiector/tablă.
5.2. de desfășurarea a seminarului/laboratorului	Laboratoare – calculatoare dotate cu: JDK 6.0, Netbeans 7.0

## 6. Competențe specifice acumulate

Competențe profesionale	<p><b>C1 Programarea în limbaje de nivel înalt</b></p> <p><b>C2 Dezvoltarea și întreținerea aplicațiilor informatice</b></p>
Competențe transversale	<ul style="list-style-type: none"> <li>- Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unor atitudini responsabile față de domeniul științific și didactic, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională</li> <li>- Desfășurarea eficientă și eficace a activităților organizate în echipă</li> <li>- Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată</li> </ul>

## 7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Însușirea noțiunilor fundamentale și a ultimelor dezvoltări din domeniul programării avansate pentru a elabora programe competitive aplicate în diverse sfere IT;</li> <li>- Deprinderea sintaxei limbajului Java și construirea de programe folosind acest limbaj</li> <li>- Dezvoltarea capacităților în vederea proiectării orientate obiect a aplicațiilor</li> </ul>
7.2 Obiectivele specifice	<p>Ob. de cunoaștere (OC): (1) să identifice structuri de date abstracte și relațiile dintre ele; (2) să descrie/identifice algoritmi pentru prelucrarea structurilor abstracte de date identificate;</p> <p>Ob. de abilitare (OAb): (1) să identifice algoritmul și structura de date adecvate unei probleme concrete; (2) să implementeze un algoritm într-un limbaj de programare de nivel înalt;</p> <p>Ob. Atitudinale (OAt): (1) să argumenteze structura structuri de date aleasă și operațiile care se pot efectua asupra ei, importanța identificării structurilor de date abstracte fiind importantă în design-ul aplicațiilor IT</p>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere în tehnici avansate de programare. Tehnica Divide et Impera. Metoda programării dinamice. Metoda Branch and Bound. Metoda Greedy. Tehnica backtracking.	Prelegere, discuție, studii de caz	2 ore
2. Tehnici de programare avansată. Algoritmi de sortare. Fire de execuție multiple (thread-uri). Fluxuri de intrare-ieșire. Serializarea datelor.	Prelegere, discuție, studii de caz	2 ore
3. Introducere în limbajul Java. Istoricul și caracteristicile limbajului Java. Portabilitatea și securitatea execuției programelor Java. Tipuri de date în limbajul Java.	Prelegere, discuție, studii de caz	2 ore
4. Java versus C++. Java versus .Net. Paradigma POO. Clase și obiecte; spații de nume și biblioteci (packages). Crearea și distrugerea obiectelor.	Prelegere, discuție, studii de caz	2 ore
5. Moștenirea și ierarhia claselor. Clase abstracte și interfețe. Tratarea excepțiilor	Prelegere, discuție, studii de caz	2 ore
6. Interfețe grafice Java (AWT și Swing). Componente grafice: containere și controale. Gestionari de poziționare a componentelor grafice.	Prelegere, discuție, studii de caz	2 ore
7. Interfețe grafice Java (AWT și Swing). Ascultarea și tratarea evenimentelor de la componentele grafice. Contexte grafice și desenarea în suprafețe de desenare (canvas).	Prelegere, discuție, studii de caz	2 ore
8. Interfețe grafice Java (AWT și Swing). Dialoguri și meniuri.	Prelegere, discuție, studii de caz	2 ore
9. Java și serviciile Internet. Programarea WEB.	Prelegere, discuție, studii de caz	2 ore
10. Client Web: applet-uri. Metodele unui applet	Prelegere, discuție, studii de caz	2 ore
11. Încărcarea claselor unui applet. Probleme de securitate la execuția unui applet. Server Web: servleți și pagini JSP.	Prelegere, discuție, studii de caz	2 ore
12. JavaBeans. Utilizarea componentelor JavaBeans în pagini JSP.	Prelegere, discuție, studii de caz	2 ore
13. Conectivitatea Java cu baze de date. Accesul la baze de date folosind JDBC.	Prelegere, discuție, studii de caz	2 ore
14. Conectivitate Java cu baze de date. Execuția unei instrucțiuni SQL. Procesarea rezultatelor. Închiderea unei conexiuni la o bază de date	Prelegere, discuție, studii de caz	2 ore
<b>8.2 Bibliografie</b>		
<ol style="list-style-type: none"> <li>1. *** <i>JDK Documentation</i>, <a href="http://java.sun.com">http://java.sun.com</a>.</li> <li>2. Bruce Eckel, <i>Thinking in Java, (3rd edition)</i>, <a href="http://www.bruceeckel.com">http://www.bruceeckel.com</a>.</li> </ol>		

3. Marty Hall, <a href="http://coreservlets.com">Core Servlets and JavaServer Pages</a> , http://coreservlets.com 4. Kathy Sierra & Bert Bates, <i>Atac la Java</i> , Ed.Teora, 2006. 5. Cristian Frăsinaru, <i>Curs Practic de Java</i> , Ed.MatrixRom, 2005.		
<b>Laborator</b>		2 ore
1. Introducere în mediul de dezvoltare Netbeans 7.0. Elemente de bază ale limbajului Java.	Studii de caz, laptop, PC, server	2 ore
2. Instrucțiuni de bază ale limbajului Java.	Studii de caz, laptop, PC, server	2 ore
3. Clase și obiecte în Java. Crearea obiectelor. Constructori. Variabile clasă	Studii de caz, laptop, PC, server	2 ore
4. Clase și obiecte în Java. Metode clasă (stative). Moștenirea.	Studii de caz, laptop, PC, server	2 ore
5. Suprascrierea metodelor în Java. Ascunderea și încapsularea datelor.	Studii de caz, laptop, PC, server	2 ore
6. Clase și metode abstracte în Java	Studii de caz, laptop, PC, server	2 ore
7. Applet-uri Java. Componentele interfeței grafice.	Studii de caz, laptop, PC, server	2 ore
8. Evenimente generate de componentele AWT	Studii de caz, laptop, PC, server	2 ore
9. Java Swing. JFrame, JApplet, JPanel, Borders	Studii de caz, laptop, PC, server	2 ore
10. Java Swing. Tabbed Panes, Scrolling Panes, Split Panes	Studii de caz, laptop, PC, server	2 ore
11. Java Swing. Etichete și butoane	Studii de caz, laptop, PC, server	2 ore
12. JList. JComboBox. JSpinner.	Studii de caz, laptop, PC, server	2 ore
13. Componente text. JTree. JTable. Meniuri. JToolBar	Studii de caz, laptop, PC, server	2 ore
14. Prezentarea proiectului individual realizat pe baza cunoștințelor dobândite la curs și laborator.	Studii de caz, laptop, PC, server	2 ore
<b>Bibliografie</b>		
1. <a href="http://netbeans.org/">http://netbeans.org/</a> - ianuarie 2018 2. <a href="http://platform.netbeans.org/tutorials/nbm-projectsamples.html">http://platform.netbeans.org/tutorials/nbm-projectsamples.html</a> -februarie 2018 3. <a href="http://www.roseindia.net/webservices/netbeans/index.shtml">http://www.roseindia.net/webservices/netbeans/index.shtml</a> -februarie 2018 4. Kathy Sierra & Bert Bates, <i>Atac la Java</i> , Ed.Teora, 2006. 5. Cristian Frăsinaru, <i>Curs Practic de Java</i> , Ed.MatrixRom, 2005.		

### 9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Conținutul este în concordanță cu structura cursurilor similare de la alte universități și acoperă aspectele fundamentale necesare familiarizării cu problematica programării orientate obiect. Abilitatea de a identifica, proiecta, implementa și analiza probleme care se pot rezolva folosind principii orientate obiect este esențială pentru orice activitate din domeniul informaticii. Competențele oferite de această disciplină sunt necesare unui specialist IT pentru a identifica soluții de rezolvare a unor probleme concrete, indiferent de domeniul specific de activitate.

### 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Însusirea cunoștințelor despre terminologia programării orientate obiect și caracteristici ale limbajului Java (OC1, OC2); Identificarea de: structuri de date abstracte, metode de rezolvare a diferite prelucrări asupra structurilor de date identificate. (OAb1, OAb2, OAt1); Capacitatea de a defini o structura abstracta de date, de a identifica relații între mai multe structuri abstracte de date și realiza prelucrări asupra rol (OAb1, OAb2, OAt1).	<i>Evaluare finală Examen scris</i>	50%
10.5 Seminar/laborator	Capacitatea de a defini o structura abstracta de date si realiza operatii simple asupra unui sir de obiecte de tipul definit. (OAb1, OAb2, OAt1).	<i>Verificare pe parcurs Proiect</i>	50%

10.6 Standard minim de performanță:

Modelarea și rezolvarea unor probleme cu grad mediu de complexitate, folosind cunoștințe de matematică și informatică.

Prezența la cursuri și seminarii conform cerințelor generale ale facultății.

- cunoașterea noțiunilor fundamentale (minim nota 5 la evaluarea finală)
- capacitatea de a aplica în practică noțiunile teoretice (minim media 5 pt. laborator)

Nota finală se calculează ca medie aritmetică a notelor acordate pentru componentele specificate la 10.4 și 10.5. Examenul se consideră promovat dacă media este cel puțin 5 (este necesar ca notele de la 10.4 și 10.5 să fie mai mari ca 5 fiecare). La fiecare dintre sesiunile de examen (inclusiv cele de reanță și măriri) nota se calculează după aceeași regulă. În sesiunea de reanță/măriri se pot susține doar probele la care nu s-a obținut notă de promovare (minim 5), cu excepția cazului în care studentul dorește să susțină și probele deja promovate.

Obs: Studenții pot participa la orele de consultații (2 module/săptămână conform planificării stabilite la începutul semestrului) în cadrul cărora titularul de curs și/sau seminar/laborator răspunde întrebărilor studenților și oferă explicații suplimentare legate de conținutul cursului, aplicațiile de la laborator și teme.

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

Data avizării în departament

Semnătura directorului de departament

Data aprobării în Consiliul Facultății

Semnătura Decanul Facultății